

AMENDMENTS TO THE CLAIMS

Please amend the Claims as follows:

1. (Original) A method for controlling concurrent access of prefix encoded nodes in a hierarchically structured document comprising steps of:
 - a. processing an explicit lock request on a node by determining ancestor nodes from said node,
 - b. deriving implicitly from said explicit lock request, a set of locks for said determined ancestor nodes,
 - c. comparing said derived set of implicit locks with existing lock modes for said determined ancestor nodes, and
 - d. granting or denying said explicit lock request on said node based on results of said comparing step.
2. (Currently Amended) The method of claim 1 ~~A method for controlling concurrent access, as per claim 1,~~ wherein said hierarchically structured document is an XML document.
3. (Currently Amended) The method of claim 1 ~~A method for controlling concurrent access, as per claim 1,~~ wherein said node is comprised of data and a node identifier (ID).
4. (Currently Amended) The method of claim 3 ~~A method for controlling concurrent access, as per claim 3,~~ wherein said explicit lock mode is any of: a shared (S), update (U), or exclusive (X) lock mode.

5. (Currently Amended) The method of claim 1 ~~A method for controlling concurrent access, as per claim 1,~~ whereupon granting said explicit lock request, one or more of said implicitly derived locks are implicitly applied to said ancestor nodes.

6. (Currently Amended) The method of claim 4 ~~A method for controlling concurrent access, as per claim 4,~~ wherein said implicitly derived lock mode is any of: an intention-shared (IS), intention-exclusive (IX), or a shared, intention-exclusive lock (SIX) mode.

7. (Currently Amended) The method of claim 6 ~~A method for controlling concurrent access, as per claim 6,~~ wherein

a. an explicit lock request on said node in lock mode S implicitly derives a set of locks in IS mode,

b. an explicit lock request on said node in lock mode X implicitly derives a set of locks in IX mode,

c. an explicit lock request on said node in lock mode IS implicitly derives a set of locks in IS mode,

d. an explicit lock request on said node in lock mode IX implicitly derives a set of locks in IX mode, and

e. an explicit lock request on said node in lock mode SIX implicitly derives a set of locks in SIX mode.

8. (Currently Amended) The method of claim 7 ~~A method for controlling concurrent access, as per claim 7~~, wherein said lock request is denied if said comparison step results in incompatibility and granted otherwise; said comparison step results in compatibility between said existing and derived lock modes if lock request mode for said node is:

- a. IS and said ancestor nodes are locked in any existing mode of: IS, IX, S, or SIX,
- b. IX and said ancestor nodes are locked in either existing mode of: IS or IX,
- c. S and said ancestor nodes are locked in either existing mode of: IS or S,
- d. SIX and said ancestor nodes are locked in existing mode of IS, and
- e. X and said ancestor nodes are not currently locked; and

said comparison step results in incompatibility between said existing and derived locked modes, otherwise.

9. (Currently Amended) The method of claim 8 ~~A method for controlling concurrent access, as per claim 8~~, wherein said comparing step is facilitated by a logical data structure indicating existing lock information for each node; said logical data structure comprising logical lock tree nodes.

10. (Currently Amended) The method of claim 9 ~~A method for controlling concurrent access, as per claim 9~~, wherein said logical lock tree nodes are comprised of at least: a node ID field, a transaction ID field, and a lock mode field.

11. (Currently Amended) The method of claim 10 ~~A method for controlling concurrent access, as per claim 10~~, whereupon granting a lock request, a logical lock tree node for said node is

created and ID of said node is inserted into said logical lock tree node ID field, a transaction ID is inserted into said logical lock tree node transaction ID field, a lock mode is inserted into said logical lock tree node lock mode field; and if logical lock tree nodes exist for said ancestor nodes, adding either one or both of: a transaction ID to said logical lock tree transaction ID fields and adding said lock mode to said logical lock tree node lock mode fields;

else

creating logical lock tree nodes for said ancestor nodes, inserting IDs of said ancestor nodes into said logical lock tree node ID fields, inserting a transaction ID into said logical lock tree node ID fields, inserting a transaction ID into said logical lock tree node transaction ID fields, and inserting a lock mode into said logical lock tree node lock mode fields.

12. (Original) An article of manufacture comprising a computer usable medium having computer readable program code embodied therein which implements concurrent access control of prefix encoded nodes in a hierarchically structured document comprising modules implementing code for:

- a. processing an explicit lock request on a node by determining ancestor nodes from said node,
- b. deriving implicitly from said explicit lock request, a set of locks for said determined ancestor nodes,
- c. comparing said derived set of implicit locks with existing lock modes for said determined ancestor nodes, and
- d. granting or denying said explicit lock request on said node based on results of said comparing step.

13. (Currently Amended) The article of manufacture of claim 12 ~~An article of manufacture for controlling concurrent access, as per claim 12,~~ wherein said hierarchically structured document is an XML document.

14. (Currently Amended) The article of manufacture of claim 12 ~~An article of manufacture for controlling concurrent access, as per claim 12,~~ wherein said node is comprised of data and a node identifier (ID).

15. (Currently Amended) The article of manufacture of claim 12 ~~An article of manufacture for controlling concurrent access, as per claim 12,~~ wherein said explicit lock mode is any of: a shared (S), update (U), or exclusive (X) lock mode.

16. (Currently Amended) The article of manufacture of claim 15 ~~An article of manufacture for controlling concurrent access, as per claim 15,~~ wherein said implicit lock mode is any of: an intention-shared (IS), intention-exclusive (IX), or a shared, intention-exclusive lock (SIX) mode.

17. (Currently Amended) The article of manufacture of claim 16 ~~An article of manufacture for controlling concurrent access, as per claim 16,~~ wherein

a. an explicit lock request on said node in lock mode S derives a set of implicit locks in IS mode,

b. an explicit lock request on said node in lock mode X derives a set of implicit locks in IX mode,

- c. an explicit lock request on said node in lock mode IS derives a set of implicit locks in IS mode,
- d. an explicit lock request on said node in lock mode IX derives a set of implicit locks in IX mode, and
- e. an explicit lock request on said node in lock mode SIX derives a set of implicit locks in SIX mode to be applied to said determined ancestor nodes.

18. (Original) The article of manufacture of claim 17 ~~An article of manufacture for controlling concurrent access, as per claim 17,~~ wherein said comparison step results in compatibility between said existing and derived lock modes if lock request mode for said node is:

- a. IS and said ancestor nodes are locked in any existing mode of: IS, IX, S, or SIX,
- b. IX and said ancestor nodes are locked in either existing mode of: IS or IX,
- c. S and said ancestor nodes are locked in either existing mode of: IS or S,
- d. SIX and said ancestor nodes are locked in existing mode of IS, and
- e. X and said ancestor nodes are not currently locked;

otherwise

said comparison step results in incompatibility between said existing and derived lock modes.

19. (Original) A system for controlling concurrent access of prefix encoded nodes in a hierarchically structured document comprising:

- a. a processor receiving as input, an explicit lock request on a node and providing as output ancestor nodes determined from said node,

b. a converter receiving as input said explicit lock request and deriving as output a set of implicit locks for said output ancestor nodes,

c. a comparator comparing said derived set of implicit locks with existing lock modes for said output ancestor nodes, and
a lock request grantor, granting or denying said explicit lock request on said node based on output of said comparator.

20. (Original) A method for controlling concurrent access of prefix encoded nodes in a hierarchically structured document comprising steps of:

a. processing an explicit lock release on a node by determining ancestors nodes from said node; said explicit lock release requested by a transaction;

b. deriving from said explicit lock release, a set of implicit lock modes for said determined ancestor nodes, and

c. releasing locks on determined ancestor nodes corresponding to said derived implicit lock mode; said locks on determined ancestor nodes originally requested by said transaction.

21. (Original) An article of manufacture comprising a computer usable medium having computer readable program code embodied therein which implements concurrent access control of prefix encoded nodes in a hierarchically structured document comprising modules executing:

a. explicit lock request processing on a node by determining ancestor nodes from said node,

b. implicit derivation of a set of locks for said determined ancestor nodes from said explicit lock request,

- c. a comparison of said derived set of implicit locks with existing lock modes for said determined ancestor nodes, and
- d. granting or denying said explicit lock request on said node based on results of said comparing step.